MSc Geospatial Sciences
Student: Adilson Pacheco
Supervisor: Prof. Tao Cheng

UCL Department of Civil, Environmental and Geomatic Engineering, Gower St, London ,WC1E 6BT

Open Source Dashboard For Crime Data Visualisation and Analysis

UCL

Introduction

Crime mapping has significantly benefited from recent technological advancements of visualisation tools. Subscription for current visualisation tools available on the market (e.g. Tableau, Power B.I and ESRI), are still relatively expensive. This means that the availability of these tools is often restricted to large law enforcement agencies and are mostly deployed in larger metropolitan areas. Open-source dashboards have emerged as the primary solution to this problem.
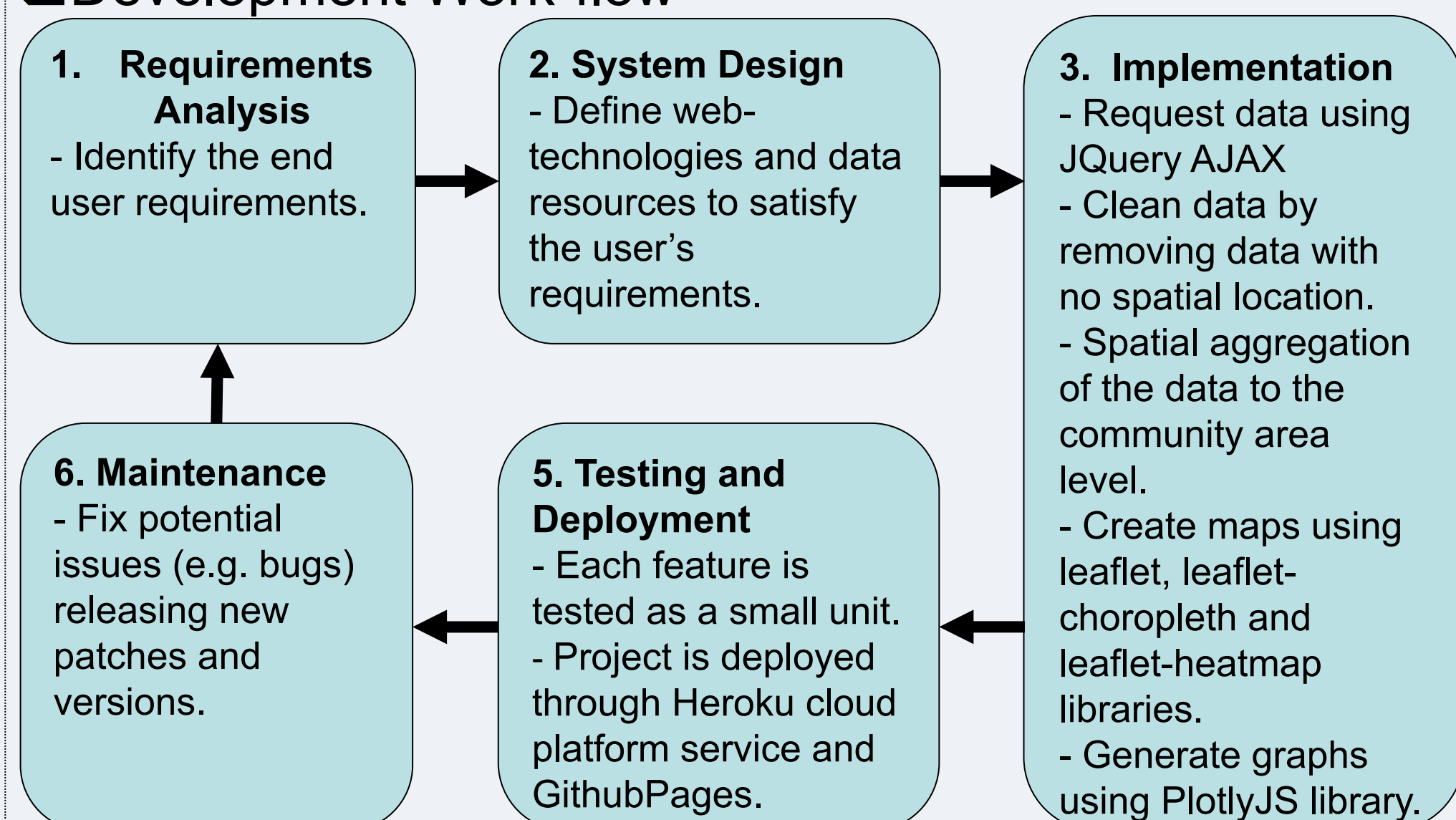
Aim and Objectives

- Develop a prototype for an open-source dashboard using JavaScript that can be deployed at any geographical location and geographical scale (e.g. borough, ward and county).
- This open-source web-based application would enable users to both visualise and analyse data dynamic and interactive using maps and charts.
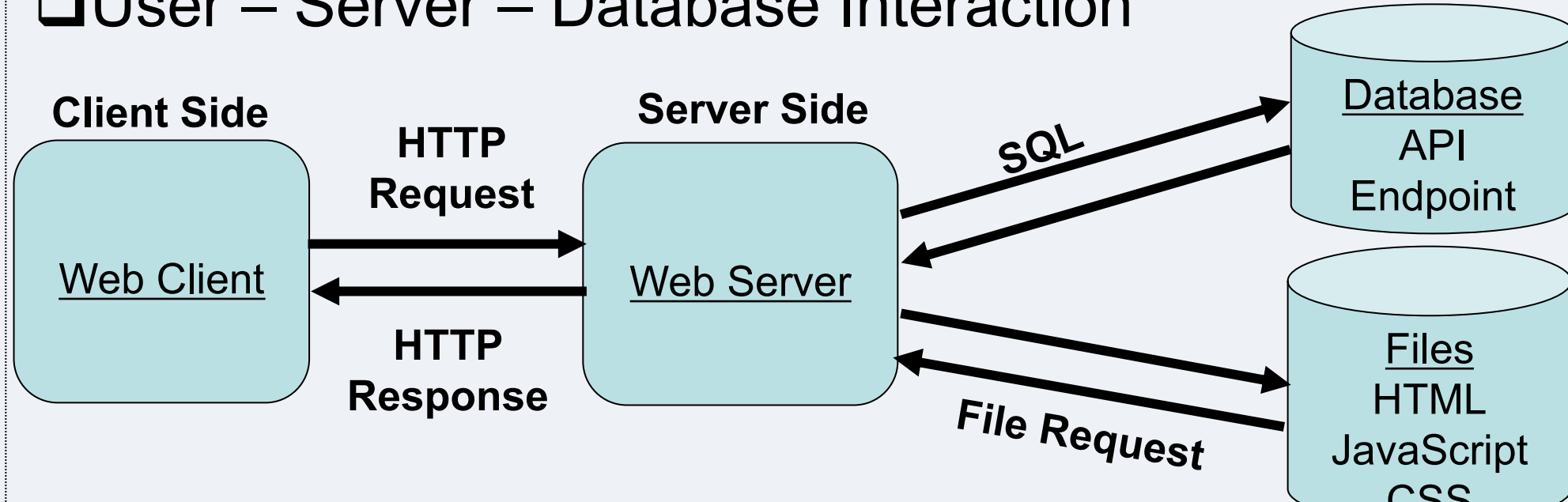
Data

1. **Chicago Crime**: API endpoint containing nearly 7 million crime records between 2001 onwards. Each incident recorded at block level containing 22 attributes (e.g. crime type, date and coordinates). (City of Chicago)[1]
2. **Chicago Community Area Boundary**: GeoJSON containing Chicago community area boundary map for the year of 2015. (Novak,2019)[2]
3. **Chicago Community Area Population**: Population size per community area based on the 2010 census. (City of Chicago)[3]

Methodology

❑ Development Work-flow

1. Requirements Analysis - Identify the end user requirements.

2. System Design - Define web-technologies and data resources to satisfy the user's requirements.

3. Implementation - Request data using JQuery AJAX - Clean data by removing data with no spatial location. - Spatial aggregation of the data to the community area level. - Create maps using leaflet, leaflet-choropleth and leaflet-heatmap libraries. - Generate graphs using PlotlyJS library.

6. Maintenance - Fix potential issues (e.g. bugs) releasing new patches and versions.

5. Testing and Deployment - Each feature is tested as a small unit. - Project is deployed through Heroku cloud platform service and GithubPages.

❑ User – Server – Database Interaction

Client Side

Web Client

HTTP Request / HTTP Response

Server Side

Web Server

SQL / File Request

Database API Endpoint

Files HTML JavaScript CSS

- User selects the filtering parameters using dropdowns in the sidebar, an AJAX request in form of HTTP request is sent to the web server.
- The web server queries the database and returns an HTTP response in a GeoJSON object.
- The HTTP response is processed and manipulated using JavaScript and displayed using HTML.
- User visualises the results on the web page (e.g. Chrome or Safari).

Results

❑ Visualisation
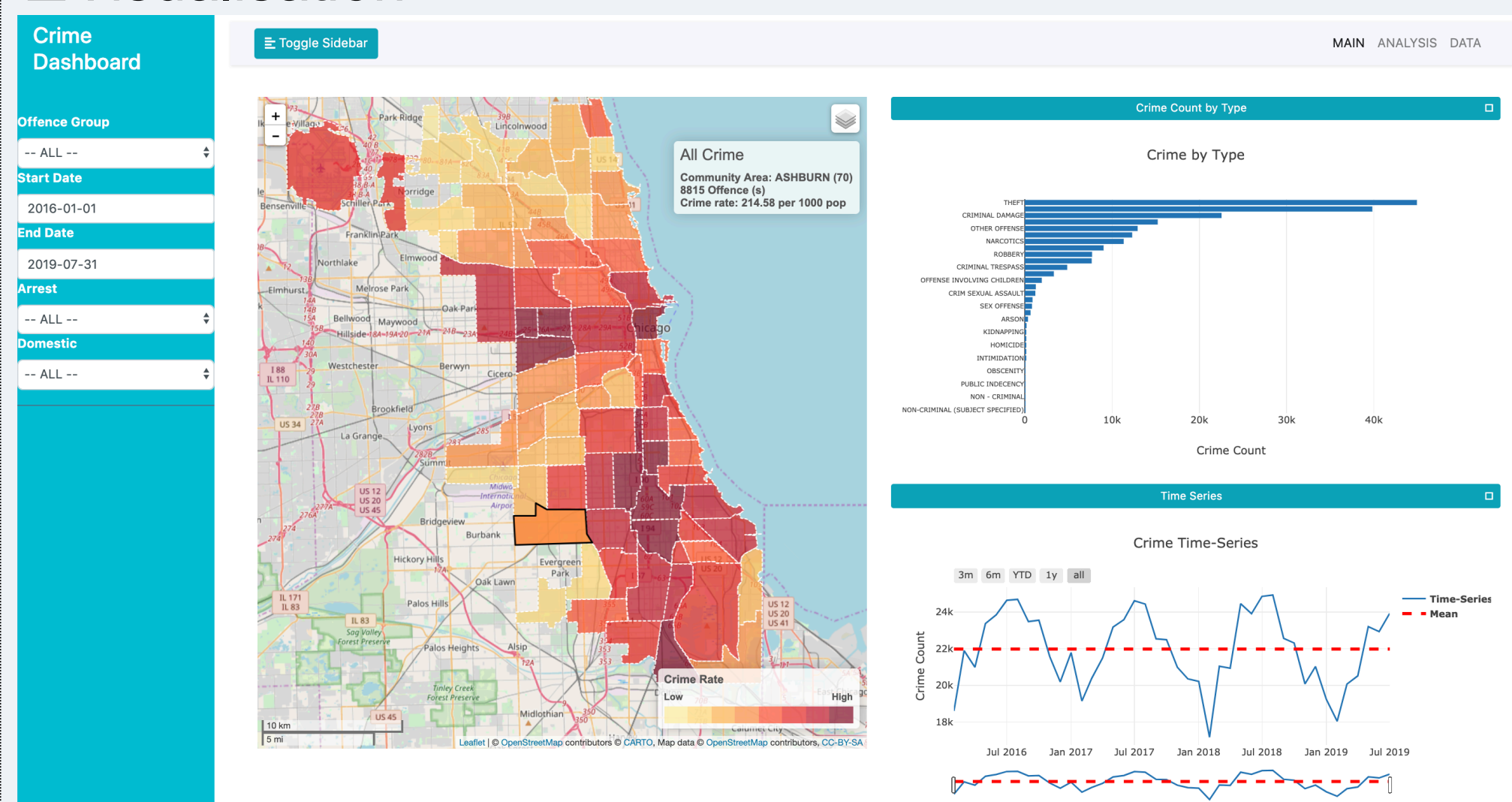
- The graphical user interface is composed of three main components, a sidebar comprising the filtering parameters, a map and collapsible buttons that display charts when clicked.
- The map is overlaid by a choropleth layer showing crime rates in different community areas of Chicago rated from low to high. The scale is defined using k-means break classification. Mouse pointing highlights the chosen community area and displays the information dynamically on the top right information box on the map.
- The charts include a display of the crime count, a histogram showing crime distribution, pie-charts comparing arrest and domestic percentage and a time-series illustrating the change in crime incidents over the months of the year.

❑ Analysis
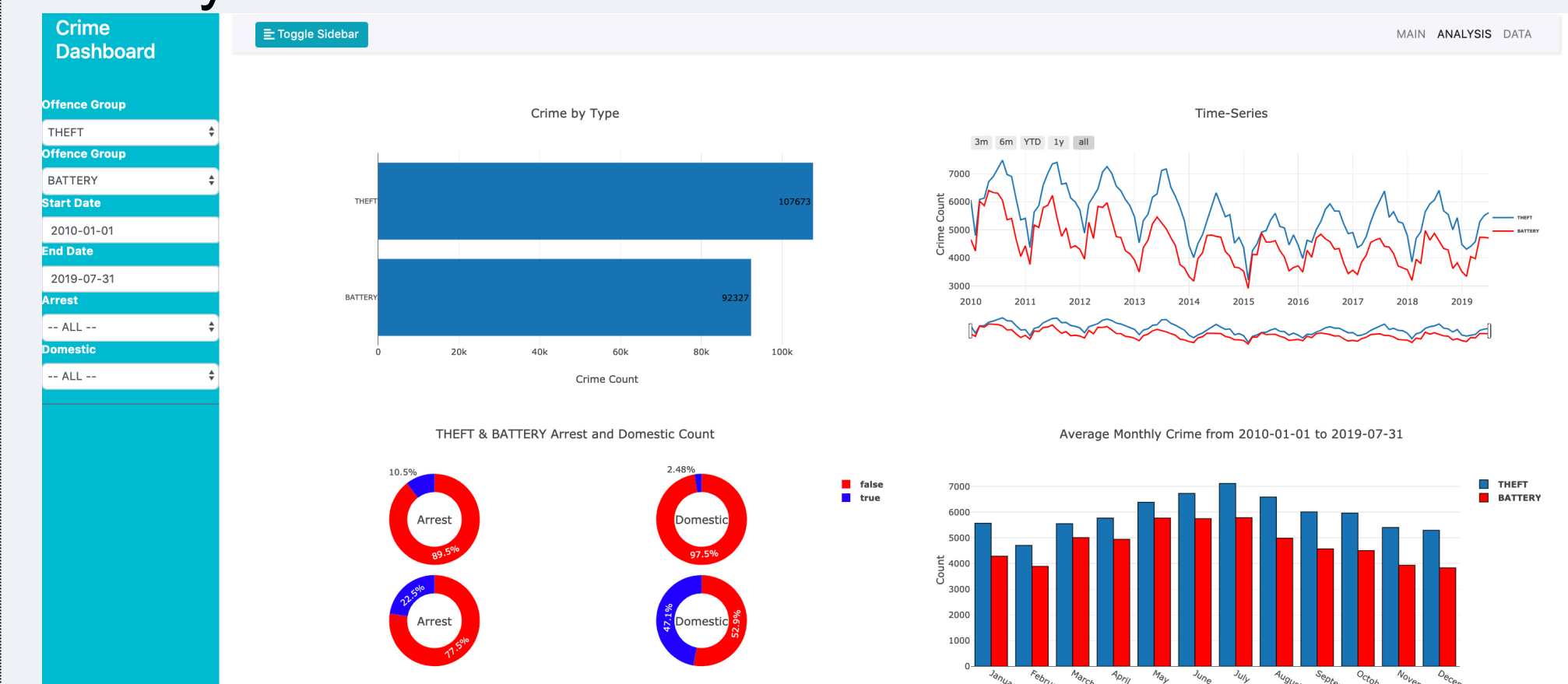
- User inputs two distinct crime types for comparative analysis of trends, distribution and rates.
- In the figure above we can see a comparative analysis between theft and battery crime. Both crimes display similar seasonality and distribution which could suggest some correlation between both crime types.

Conclusion

- Open-source applications are well suited to solve small-scale problems providing solutions that are tailored to a specific problem while proprietary tools generally handle bigger problems.
- Open-source tools require significant time effort and individuals with a high level of coding which contrasts to using proprietary software.

❑ Future work

- Carry out further user testing to detect bugs more efficiently to fulfil the maintenance stage of the development.
- Improve crossfilter functionality between the map layers and the graphs for smoother interactivity and better drill-down to the data.
- Improve spatial aggregation of the data to the smallest spatial unit available to enhance precision.
- Introduce spatio-temporal visualisation techniques for more robust analysis, for example, 3D space-time Kernel density estimation.

References

1. Anon 2019. *Crimes - 2001 to present | City of Chicago | Data Portal*. [online] Chicago. Available at: <https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2/data> [Accessed 28 Aug. 2019].
2. Novak, T., 2019. *Exploring public Chicago crimes data set in Python: RandomFractals/ChicagoCrimes*. [Jupyter Notebook] Available at: <https://github.com/RandomFractals/ChicagoCrimes> [Accessed 28 Aug. 2019].
3. Anon 2019. *Community Area 2000 and 2010 Census Population Comparisons*. [online] Available at: <https://www.chicago.gov/content/city/en/depts/dcd/supp_info/community_area_2000and2010censuspopulationcomparisons.html> [Accessed 28 Aug. 2019].

SpaceTimeLab for Big Data Analytics